REMARKS

The above Amendments and these Remarks are in reply to the Office Action mailed October 31, 2006.

Currently, claims 1-37 are pending. Applicants have amended claims 4 and 5. Applicants respectfully request reconsideration of claims 1-37.

I. Summary of Amendments

Claims 4 and 5 have been amended to correct the grammar of the claims.

II. Rejection of Claims 1-5, 7, 13-15, 18, 19, 21, 26, 27, 29, and 35 Under 35 U.S.C. 102(b)

Claims 1-5, 7, 13-15, 18, 19, 21, 26, 27, 29, and 35 have been rejected under 35 U.S.C. 102(b) as being anticipated by the Steele reference. Applicants respectfully traverse these rejections on the basis that Steele does not disclose all of the limitations of the claims.

Claim 1 discloses code that "includes one or more expressions and one or more markers that specify when said one or more expressions should be evaluated during execution of said program." The Examiner argues that Steele discloses code with these limitations, yet no where in Steele is this expressly stated or suggested. The `eval-when` form referred to in Steele "allows pieces of code to be executed only at compile time, only at load time, or when interpreted but not compiled," (p. 11, line 8) but only in these certain situations (`compile`, `load`, or `eval`). These situations occur prior to execution time, not during execution as is recited in claim 1. For example, Steele discloses the following:

> `eval` specifies that the interpreter should process the body. `compile` specifies that the compiler should evaluate the body at compile time in the compilation context. `load` specifies that the compiler should arrange

> to evaluate the forms in the body when the compiled file containing the `eval-when` form is loaded (p.11, lines 14-17).

These situations do not specify when code should be evaluated **during** execution, as Applicants' invention recites in claim 1.

Additionally, Steele does not disclose additional functionality that "evaluates said one or more expressions during execution of said program at one or more times specified by said one or more markers," as is recited in claim 1. The Examiner argues that Steele's interpreter and compiler process the body according to the situations specified (p.11, line 14), yet no where in the reference is it stated or implied that the additional functionality is evaluated **during** execution at times specified by the situations. Rather, Steele teaches to perform the function when compiling, loading or interpreting, all of which are before execution in Steele. Because Steele does not disclose the above-quoted limitations, Applicants assert that claim 1, as well as claims 2-5, 7, 13-15, 21, and 29, are in condition for allowance.

Furthermore, claim 5 discloses that the "step of automatically providing additional functionality to said code includes adding code that creates an object for each constraint, adds functions to said object that set said variables, and adds functions that set dependencies for said expressions." No where in the Steele reference does it disclose that providing additional functionality to the code includes adding code that adds functions that set dependencies for the expressions. Instead Steel discloses a "call to `set-macro-character` to be executed in the compiler's execution environment, thereby modifying its reader syntax table" (p. 12, lines 1-5). Yet no dependencies are set by the additional functionality, as is claimed in Applicants' invention.

Regarding claim 18, Steele does not disclose "an expression defining a first variable, said expression is dependent on a changeable item." A changeable item is described in the specification to be a variable, property, etc. upon which an expression is dependent. "Additional functionality is provided that evaluates the expression when the item changes in order to update the first variable" (p. 5, paragraph [0016]). The Examiner argues that on page 14, line 14 Steele discloses “a first variable ‘x’ and expression,” yet there is no changeable item within the mentioned expression. Page 14, line 14 of the Steele reference shows the following:

```
(defun bar (x) (defun foo () (+x 3)))
```

In this case, the expression is not dependent on a changeable item within this line of code. Instead, the expression is dependent on the first variable itself. Consequently, in the Steele code, the first variable cannot be updated by the expression as claimed in the present invention because there is no changeable item. Because Steele does not disclose an expression being dependent on a changeable item, as suggested by the Examiner, Applicants assert that claim 18, as well as claims 19, 21, 26, 27, 29, and 35, are in condition for allowance.

III. Rejection of Claims 6, 8-12, 16, 17, 20, 22-25, 28, 30-34, 36, and 37 Under 35 U.S.C. 103(a)

A. Steele in view of Rodriguez

Claims 6, 8, 16, 30, and 36 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Steele in view of Rodriguez. Because the cited prior art, alone or in combination, does not teach all of the limitations of the rejected claims, Applicants assert that the claims are in condition for allowance.

Regarding claims 6, 8, and 16, neither Steele nor Rodriguez discloses code that specifies when expressions should be evaluated **during** execution of a program. Concerning claims 30 and 36, neither Steele nor Rodriguez discloses an expression being dependent on a changeable item. Instead, the Rodriguez reference discusses using Lisp scripts within HTML code for interactive applications over the Internet. Therefore, the combination of the Steele and Rodriguez references do not disclose all of the limitations of claims 6, 8, 16, 30, and 36.

B. Steele

Claims 9-12, 22, 23, 25, 31, 32, 34 have been rejected under 35 U.S.C. 103(a) as being obvious over the Steele reference. Because the cited prior art does not teach or suggest all of the limitations of the rejected claims, Applicants assert that the claims are in condition for allowance.

Claims 9-12, 25, and 34 include the limitation of code that specifies when expressions should be evaluated **during** execution of a program. As discussed above, Steele does not disclose, teach, or suggest this limitation. Therefore, claims 9-12, 25, and 34 are not obvious over Steele.

Claims 22, 23, 25, 31, 32, and 34 include the limitation of an expression being dependent on a changeable item or dependency, the dependency being changeable as well. As described above, Steele does not disclose, teach, or suggest this limitation. Therefore, claims 22, 23, 25, 31, 32, and 34 are not obvious over Steele.

C. <u>Steele in view of Hickey</u>

Claim 17 has been rejected under 35 U.S.C. 103(a) as being unpatentable over Steele in view of Hickey. Because the cited prior art, alone or in combination, does not teach all of the limitations of the rejected claims, Applicants assert that the claims are in condition for allowance.

Neither Steele nor Hickey discloses code that specifies when expressions should be evaluated **during** execution of a program. Instead, the Hickey reference discloses Lisp as a tool for web programming. Therefore, the combination of the Steele and Hickey references do not disclose all of the limitations of claim 17.

D. <u>Steele in view of Haible</u>

Claims 20, 24, 28, and 33 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Steele in view of Haible. Because the cited prior art, alone or in combination, does not teach all of the limitations of the rejected claims, Applicants assert that the claims are in condition for allowance.

Neither Steele nor Haible discloses an expression being dependent on a changeable item or dependency, the dependency being changeable as well. Instead, the Haible reference discloses implementations for Common Lisp. Therefore, the combination of the Steele and Haible references do not disclose all of the limitations of claims 20, 24, 28, and 33.

E. Steele in view of Rodriguez and Haible

Claim 37 has been rejected under 35 U.S.C. 103(a) as being unpatentable over Steele in view of Rodriguez and further in view of Haible. Because the cited prior art, alone or in combination, does not teach all of the limitations of the rejected claims, Applicants assert that the claims are in condition for allowance.

Steele, Rodriguez, and Haible do not disclose an expression being dependent on a changeable item. Therefore, the combination of these references does not disclose all of the limitations of claim 37.

Based on the above amendments and these remarks, reconsideration of claims 1-37 is respectfully requested.

The Examiner's prompt attention to this matter is greatly appreciated. Should further questions remain, the Examiner is invited to contact the undersigned agent by telephone.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 501826 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: January 31, 2007

By: /Michelle Esteban/
Michelle Esteban
Reg. No. 59,880

VIERRA MAGEN MARCUS & DENIRO LLP
575 Market Street, Suite 2500
San Francisco, California 94105-4206
Telephone: (415) 369-9660
Facsimile: (415) 369-9665